ab 13. November 2024
Prof. A. Scrinzi
T. Halenke
A. Limaco

Einführung ins Programmieren
WS 24/25
Übungsblatt 4

4.1: Code lesen: Binomialkoeffizienten

Für die Binomialkoeffizienten gilt die folgende Rekursionsformel

$$\binom{n+1}{k-1} = \binom{n}{k-1} + \binom{n}{k} \tag{1}$$

mit $k = 1, \ldots, n$ und $\binom{n}{0} = \binom{n}{n} = 1$, mit Beginn bei $n = 1$
Der folgende Code berechnet die Koeffizienten nach obiger Formel. Erklären Sie **jede einzelne Zeile** des Codes mit einem kurzen Kommentar, entweder auf Papier oder direkt im Code (siehe `Tutorials/blatt04_1.cpp` am git bzw. Moodle).

```
#include <iostream>
#include <vector>
int main(){
    std::cout<<"compute binomial coefficients for n, enter n: ";
    int nFinal;
    std::cin>>nFinal;
    if(0>nFinal or nFinal>10){
        std::cout<<"need 0<n and limited to n<=10, got: "
                 <<nFinal<<std::endl;
        exit(0);
    }
    std::vector<int> binomN(2,1);
    for(int n=1;n<nFinal;n++){
        std::vector<int> next(1,1);
        for(size_t k=1;k<binomN.size();k++){
            next.push_back(binomN[k-1]+binomN[k]);
        }
        next.push_back(1);
        binomN=next;
    }
    std::cout<<"Binomial coefficients for n="<<nFinal<<std::endl;
    for(auto n: binomN)std::cout<<" "<<n;
    std::cout<<std::endl;
}
```

4.2: Gebrauch von `std::vector`

Vektoren im mathematischen Sinn sollten Addition und Multiplikation mit einem Skalar erlauben, also z.B.

$$\vec{c} = \vec{a} + \alpha\vec{b}, \qquad \alpha \in \mathbb{R}.$$

Die `std::vector` sind keine Vektoren im mathematischen Sinn, sondern bloss "Container". Schreiben Sie einen Code, der die obige Operation für `std::vector` ausführt, gemäss der folgenden Skizze:

```
int main() {
// create two std::vector's a and b
// a: 10,9,8,7,6,5,4
// b: 1.1,1.2,5.7,0.12,0.00001, then 0 until size() matches the a.size()
// use types (integer or decimal) that are best suited for the task

// get a real number alfa from input

// write a loop for c = a + alfa * b with a suitable type for c

// output the coefficients of c into a single line

// comment your code, explain your choices
}
```

4.3: Gebrauch von `std::map`

Wir wollen alle Zahlen $< N$ auflisten, die die Primzahlen 2,3,5,7,11 als Faktoren enthalten. Das Resultat soll so aussehen (Ausgabe teileweise verkürzt):

```
 p  is factor of...
2: 2 4 6 8 10 12 14 ... 116 118 120
3: 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 ... 114 117 120
5: 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 ... 115 120
7: 7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119
11: 11 22 33 44 55 66 77 88 99 110
```

Ergänzen Sie den folgenden Code, um obiges Resultat zu erhalten:

```
int main() {
std::vector<int> primes({2,3,5,7,11});
std::map<int,std::vector<int>> factorizes;
for(int n=2;n<...;n++){
    for(auto p: primes){
        if(n%p==0)....;
    }
}
std::cout<<" p  is factor of..."<<std::endl;
for(auto p: factorizes){
    std::cout<<p.first<<":";
    for(auto n: p....)std::cout<<" "<<n;
    std::cout<<std::endl;
```

```
    }
}
```